UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/669,948	09/24/2003	David Dice	6000-32500	8314

58467 7590 06/26/2009
MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER	
ANYA, CHARLES E	

ART UNIT	PAPER NUMBER
2194	

NOTIFICATION DATE	DELIVERY MODE
06/26/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patent_docketing@intprop.com
ptomhkkg@gmail.com

Office Action Summary	**Application No.** 10/669,948	**Applicant(s)** DICE ET AL.	
	Examiner CHARLES E. ANYA	**Art Unit** 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 September 2003*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-73* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-73* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *See Continuation Sheet*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :7/19/04; 12/22/05; 1/3/06; 1/30/06.

DETAILED ACTION

1. Claims 1-73 are pending in this application.

Claim Objections

2. Claims 4-6, 20-24, 31, 35, 36, 42 and 43, 68-71 are objected to because of the following informalities:

Claims 4-6, 20-24, 31, 35, 36, 42 and 43 seem to include typographical error. Specifically, the term "the lock" on line 2 of claims 4-6, 20-24, 31, 35, and 42 and line 4 of claims 36 and 43 seem to have been used in error.

For the purpose of this office action the Examiner would replace "the lock" with "the biasable lock".

Claim 68 seems to include typographical error. Specifically, the term "either a single-threaded and" on line 2 of claim 68 seems to have been used in error.

For the purpose of this office action the Examiner would replace "either a single-threaded and" with "either a single-threaded **or**".

Claims 69-71 seem to include typographical error. Specifically, the term "claim 62" on line 1 of claims 69-71 seems to have been used in error.

For the purpose of this office action the Examiner would replace "claim 62" with "claim 68".

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 43, 44, 58-67 and 72-73 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The term "extremely" in claims 43 and 44, “substantially less than” in claim 58 and "potentially" in claim 72 are a relative term which renders the claim indefinite. These terms "extremely", “substantially less than” and "potentially" are not defined by the claim, and after a text of the specification for these terms, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

For the purpose of this office action the Examiner would broadly interpret the terms since there are not defined or disclosed.

4. As to claims 59-67, they are rejected for the same reason as claim 58 above.

5. As to claim 73, it rejected for the same reason as claim 72 above.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement

> thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 1-35, 46-57, 62-67, 71 and 73 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 1, 31, 46 and 62 are directed to a “mutual exclusion mechanism”, “biasable lock”, “lock” and “computer program product” respectively. The body/structure of the claim 1 indicates that the “mutual exclusion mechanism” include “a biasable lock” and “acquisition and release sequence”; claim 31 indicates that the “biasable lock” include “a fast path acquisition sequence" and "a second acquisition sequence"; claim 46 indicates that the "lock" include "lock state" and "bias state" and claim 62 indicates that the “computer program product” includes “instructions”.

These components of claims 1, 31, 46 and 62 are software per se, as a result the claimed mutual exclusion mechanism”, “biasable lock”, “lock” and “computer program product” are software per se.

The claimed “mutual exclusion mechanism”, “biasable lock”, “lock” and “computer program product” is therefore not process, a machine, a manufacture or a composition of matter and as such not directed to statutory subject matter.

7. As to claims 2-35, 47-51 and 63-67, they are rejected for the same reason as claims 1, 31, 46 and 62 above.

8. Claim 52 is directed to non-statutory subject matter because the claimed "computer readable medium" is not limited to storage medium. Although not disclosed, one of ordinary skill in the art at the time the invention could interpret the claimed "computer readable medium" to include transmission media (e.g. signal bearing media, carrier wave, wireline, communication medium etc.). As a transmission medium, the claimed "computer readable medium" is directed to non-statutory subject matter.

9. Claims 53 – 57, they are rejected for the same reason as claim 52 above.

10. Claims 67, 71 and 73 are directed to non-statutory subject matter because the claimed "computer readable medium" is not limited to storage medium.

According to claims 67, 71 and 73, the claimed computer readable medium is not limited storage medium, instead it is claimed to include both storage media (e.g. disk, tape etc) and transmission media (e.g. network, wireline, wireless etc.). Claims directed to transmission medium are directed to non-statutory subject matter, thus claims 67, 71 and 73 are directed to non-statutory subject matter.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the

> applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

11. Claims 1, 3 , 6, 24, 31, 32, 35, 36, 42-44, 46-48, 50, 52, 53, 57-59 and 62-67 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Pat. No. 6,662,364 B1 issued to Burrows et al.

12. As to claim 1, Burrows teaches a mutual exclusion mechanism comprising:

a biasable lock ("...target mutex..." Col. 2 Ln. 53 - 67, "…Mutex M…" Col. 4 Ln. 1 – 18, Ln. 56 – 67, "…Mutex M (210) Col. 5 Ln. 1 – 36); and

acquisition and release sequences ("…acquire mutex M(210)…releases the mutex..." Col. 5 Ln. 1 - 14) that, when executed by a thread to which bias has been directed, are free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, Col. 4 Ln. 1 – 18, Ln. 56 – 67).

13. As to claim 3, Burrows teaches the mutual exclusion mechanism of claim 1, wherein the acquisition and release sequences include only read and write operations when executed by the thread to which bias has been directed ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 67).

14. As to claim 6, Burrows teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock is biased on acquisition ("…the mutex is updated to indicate the new state of the mutex…" Col. 5 Ln. 29 – 35).

15. As to claim 24, Burrows teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock is rebiasable to another thread during course of a computation that employs the biasable lock (Step 280 Col. 5 Ln. 37 – 54).

16. As to claim 31, Burrows teaches a biasable lock that provides at least two acquisition sequences, a fast path acquisition sequence for a thread to which the biasable lock has been biased ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67) and a second acquisition sequence, the fast path acquisition sequence optimized with respect to the second acquisition sequence ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4).

17. As to Claim 32, Burrows teaches the biasable lock of claim 31, wherein the fast path acquisition sequence is free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic

synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

18. As to claim 35, Burrows teaches the biasable lock of claim 31, wherein the biasable lock is further rebiasable ("…Treq is associated with Mutex M…, the request is granted…" Col. 5 Ln. 37 – 42).

19. As to claim 36, Burrows teaches a method of providing an efficient locking mechanism in program code, the method comprising:

a instantiating biasable lock ("…target mutex..." Col. 2 Ln. 53 - 67, Col. 3 Ln. 1 – 35, Col. 1 – 18, Mutexes 128 Col. 4 Ln. 46 – 48, "…new created mutexes…" Col. 7 Ln. 41 – 52); and

for a thread to which bias has been directed, releasing and acquiring the biasable lock using fast path instruction sequences that are free of atomic read- modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

20. As to claim 42, Burrows teaches the method of claim 36, further comprising: rebiasing the lock to another thread (Step 280 Col. 5 Ln. 37 – 54).

21. As to claim 43, Burrows teaches the method of claim 36, further comprising: executing the program code as a multi-threaded application ("…multithreaded…" Col. 2 Ln. 30 – 49), the biasable lock allowing a single thread of the executing program code to repeatedly acquire and release the lock with extremely low overhead ("…Treq is associated with Mutex M…, the request is granted…" Col. 5 Ln. 37 – 42).

22. As to claim 44, Burrows teaches the method of claim 43, after rebiasing to another thread, allowing the another thread to repeatedly acquire and release the lock with extremely low overhead (Step 280 Col. 5 Ln. 37 – 54).

23. As to claim 46, Burrows teaches a lock that maintains both a lock state ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4) and bias state ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67), whereby acquisition of the lock by a thread to which bias has been directed is more efficient than acquisition of the lock by another thread ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4).

24. As to claim 47, Burrows teaches the lock of claim 46, wherein the lock is rebiasable to another thread during course of a computation that employs the lock (Step 280 Col. 5 Ln. 37 – 54).

25. As to claim 48, Burrows teaches the lock of claim 46, including acquisition and release sequences that, when executed by the thread to which bias has been directed, are free of atomic read- modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

26. As to claim 50, Burrows teaches the lock of claim 48, wherein the acquisition and release sequences include only read and write operations when executed by the thread to which bias has been directed ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 67).

27. As to claim 52, Burrows teaches a computer program product including a mutual exclusion mechanism embodied therein, the computer program product embodied in a computer readable medium and comprising:

a data structure instantiable in memory (Disk 104/System Memory Unit 114) of a processor (Central Processing Unit 102) to implement a lock that includes a bias attribute ("…flag…" Col. 4 Ln. 46 – 48); and a lock acquisition sequence of operations executable by the processor, the lock acquisition sequence having a fast path for a thread to which bias has been directed ("...fast nonatomic load/store sequence..." Col. 2 Ln. 53 – 64, "...fast nonatomic synchronization sequence..." Col. 3 Ln. 1 – 3, "...fast nonatomic sequence, H(M)=0..." Col. 4 Ln. 1 – 18, Ln. 56 – 67) and a second path ("...more complex procedure for synchronization..." Col. 3 Ln. 1 – 3, "...expensive atomic hardware instructions, H(M0 = 1..." Col. 4 Ln. 1 – 4), the fast path optimized with respect to the second path ("...more complex procedure for synchronization..." Col. 3 Ln. 1 – 3, "...expensive atomic hardware instructions, H(M0 = 1..." Col. 4 Ln. 1 – 4).

28. As to claim 53, Burrows teaches the computer program product of claim 52, wherein, when executed by the thread to which bias has been directed, the lock acquisition sequence is free of atomic read-modify-write operations ("...fast nonatomic load/store sequence..." Col. 2 Ln. 53 – 64, "...fast nonatomic synchronization sequence..." Col. 3 Ln. 1 – 3, "...fast nonatomic sequence, H(M)=0..." Col. 4 Ln. 1 – 18, Ln. 56 – 67).

29. As to claim 57, Burrows teaches the computer program product of claim 52, wherein the lock is rebiasable to another thread during course of a computation that employs the lock (Step 280 Col. 5 Ln. 37 – 54).

30. As to claim 58, Burrows teaches a software method comprising:

biasing a lock to a first thread of execution ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67); and

subsequently acquiring the lock for, or by, the first thread with computational overhead substantially less than for a second thread to which the lock is not currently biased ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4).

31. As to claim 59, Burrows teaches the software method of claim 58, wherein the lock acquiring, when performed by, or for, the first thread, is free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

32. As to claim 62, Burrows teaches a computer program product encoding instructions that implement a biasable lock ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

33. As to claim 63, Burrows teaches the computer program product of claim 62, further comprising: a biasing sequence ("…fast nonatomic load/store sequence…" Col.

2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

34. As to claim 64, Burrows teaches the computer program product of claim 62, further comprising: a lock acquisition sequence that when executed by a thread to which bias has been directed, is free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

35. As to claim 65, Burrows teaches the computer program product of claim 62, further comprising: a lock release sequence that when executed by a thread to which bias has been directed, is free of atomic read-modify-write operations ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

36. As to claim 66, Burrows teaches the computer program product of claim 62, further comprising: a rebiasing sequence ("…Treq is associated with Mutex M…, the request is granted…" Col. 5 Ln. 37 – 42).

37. As to claim 67, Burrows teaches the computer program product of claim 62, embodied in at least one computer readable medium selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium (Main Non-Volatile Storage Unit 102 Col. 4 Ln. 25 – 30).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

38. Claims 2, 23, 33, 49, 54, 60 and 61 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. In view of U.S. Pat. No. 6,757,891 B1 issued to Azagury et al.

39. As to claim 2, Burrows is silent with reference to the mutual exclusion mechanism of claim 1, wherein the acquisition and release sequences are further free of memory barrier operations, at least when executed by the thread to which bias has been directed.

Azagury teaches the mutual exclusion mechanism of claim 1, wherein the acquisition and release sequences are further free of memory barrier operations, at least when executed by the thread to which bias has been directed (figure 6 "…does not use memory synchronization operations (e.g. sync on PowerPC)…" Col. 8 Ln. 4 – 11).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Azagury because teaching of Azagury would improve the system of Burrows by ensuring correct program operation on a multi-processor that is not sequentially consistent in order to avoid unnecessary memory synchronization costs (Azabury Col. 7 Ln. 54 – 64).

40. As to claim 23, Azagury teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock includes a monitor provided by a Java virtual machine implementation ("…monitors…" Col. 3 Ln. 32 – 45, "…monitor enter and monitor exit…" Col. 4 Ln. 26 – 49), the monitor augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed ("…monitor enter without synchronization…monitor exit without synchronization…" Col. 4 Ln. 26 – 49).

41. As to claim 33, Azagury teaches the biasable lock of claim 31, wherein the fast path acquisition sequence is free of memory barrier operations (figure 6 "…does not use memory synchronization operations (e.g. sync on PowerPC)…" Col. 8 Ln. 4 – 11).

42. As to claim 49, Azagury teaches the lock of claim 48, wherein the acquisition and release sequences are further free of memory barrier operations, at least when executed by the thread to which bias has been directed (figure 6 "...does not use memory synchronization operations (figure 6 "...does not use memory synchronization operations (e.g. sync on PowerPC)..." Col. 8 Ln. 4 – 11).

43. As to clam 54, Azagury teaches the computer program product of claim 53, wherein the acquisition sequence is further free of memory barrier operations, at least when executed by the thread to which bias has been directed (figure 6 "...does not use memory synchronization operations (e.g. sync on PowerPC)..." Col. 8 Ln. 4 – 11).

44. As to claim 60, Azagury teaches the software method of claim 58, wherein the lock acquiring, when performed by, or for, the first thread, is further free of memory barrier operations (figure 6 "...does not use memory synchronization operations (e.g. sync on PowerPC)..." Col. 8 Ln. 4 – 11).

45. As to claim 61, Azagury teaches the method of claim 58, further comprising: rebiasing the lock to a third thread of execution ("...other threads..." Col. 6 Ln. 65 – 67, Col. 7 Ln. 1 – 13).

46. Claims 4, 5, 7, 25, 26, 34, 37-41 and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. In view of U.S. Pat. No. 6,772,153 B1 issued to Bacon et al.

47. As to claim 4, Burrows is silent with reference to the mutual exclusion mechanism of claim 1, wherein the biasable lock is initially unbiased.

Bacon teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock is initially unbiased ("…Data 20C…" Col. 3 Ln. 10 – 37, Col. 4 Ln. 39 – 47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Bacon because teaching of Bacon would improve the system of Burrows by providing a mechanism for assigning a new lock on request or dynamically.

48. As to claim 5, Burrows is silent with reference to the mutual exclusion mechanism of claim 1, wherein the biasable lock is biased on creation.

Bacon teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock is biased on creation ("…any thread 6 can obtain a new system lock by calling a function createSystemLock…" Col. 4 Ln. 25 – 31, Step 502 Col. 6 Ln. 1 – 9).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Bacon because teaching of Bacon would improve the system of Burrows by providing a function mechanism for obtaining a new lock when needed (Bacon Col. 4 Ln. 25 – 28).

49. As to claim 7, Bacon teaches the mutual exclusion mechanism of claim 1, wherein the bias is directed to a thread other than a creating thread ("…createSystemLock…" Col. 4 Ln. 25 – 31).

50. As to claim 25, Bacon teaches the mutual exclusion mechanism of claim 1, wherein revocation of bias by, or on behalf of, a contending thread, is mediated, at least in part, using a signal handler ("…Locked Flag 20B is set to zero…" Col. 3 Ln. 1 – 12).

51. As to claim 26, Bacon teaches the mutual exclusion mechanism of claim 1, wherein revocation of bias by, or on behalf of, a contending thread, is mediated, at least in part, using a cross-call ("…systemRelease…" Col. 4 Ln. 15 – 18).

52. As to claim 34, Bacon teaches the biasable lock of claim 31, wherein the second acquisition sequence implements one of an MCS lock, a TATAS lock, a lock consistent with that provided by a POSIX pthread Mutex library and a Java monitor ("…POSIX…" Col. 4 Ln. 25 – 37).

53. As to claim 37, Bacon teaches the method of claim 36, further comprising: directing the bias to the thread coincident with a first acquisition of the biasable lock ("…any thread 6 can obtain a new system lock by calling a function createSystemLock…" Col. 4 Ln. 25 – 31, Step 502 Col. 6 Ln. 1 – 9).

54. As to claim 38, Bacon teaches the method of claim 36, further comprising: directing the bias to the thread coincident with the instantiation ("…any thread 6 can obtain a new system lock by calling a function createSystemLock…" Col. 4 Ln. 25 – 31, Step 502 Col. 6 Ln. 1 – 9).

55. As to claim 39, Bacon teaches the method of claim 36, further comprising: directing the bias to the thread coincident with creation of an object ("…When a thread T creates an object O, the system sets O to be unlocked, non-shared, and owned by T…" Col. 4 Ln. 60 – 67).

56. As to claim 40, Bacon teaches the method of claim 36, wherein directing of bias to the thread is performed by another thread ("…createSystemLock…" Col. 4 Ln. 25 – 31).

57. As to claim 41, Burrows teaches the method of claim 36, further comprising: for a thread other than the thread to which bias has been directed, acquiring the biasable lock using an instruction sequence that unbiases the lock, if then biased ("…more complex procedure for synchronization…" Col. 3 Ln. 1 – 3, "…or by conventional expensive atomic hardware instructions, H(M0 = 1…" Col. 4 Ln. 1 – 4).

58. As to claim 55, Burrows teaches the computer program product of claim 52, further comprising: a lock release sequence of operations executable by the processor ("…the thread holding mutex M releases the mutex…" Col. 5 Ln. 1 – 11) and **Bacon teaches** the lock release sequence having a fast path for the thread to which bias has been directed ("…acquire and release a lock without using atomic operations…" Col. 1 Ln. 39 – 65, Col. 4 Ln. 66 – 67) and a second path ("…changes the state of the lock such that future lock/unlock operations use atomic…" Col. 2 Ln. 1 – 10), the fast path optimized with respect to the second path ("…without requiring any atomic operations and/or the use of expensive synchronization primitives…" Col. 2 Ln. 1 – 10).

59. Claims 8, 11-13, 17, 18, 51 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. In view of U.S. Pat. No. 6,430,649 B1 issued to Chaudhry et al.

60. As to claim 8, Burrows is silent with reference to the mutual exclusion mechanism of claim 1, wherein the acquisition sequence employs a programming construct that precludes reordering of a particular read before a particular write.

Chaudhry teaches the mutual exclusion mechanism of claim 1, wherein the acquisition sequence employs a programming construct that precludes reordering of a particular read before a particular write ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Chaudhry because teaching of Chaudhry would improve the system of Burrows by ensuring that a particular read operation does not overtake a preceding write operation by flushing a write buffer in a load store unit before the read operation takes place including enforcing dependencies between memory references without incurring the delays inherent in member operations (Chaudhry Col. 1 Ln. 50 – 57, Col. 2 Ln. 1 – 3).

61. As to claim 11, Chaudhry teaches the mutual exclusion mechanism of claim 8, wherein the programming construct employs collocation of the target of the particular read and the target of the particular write ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

62. As to claim 12, Chaudhry teaches the mutual exclusion mechanism of claim 8, wherein the programming construct employs a memory barrier interposed between the particular read and the particular write ("…insert a "membar"…" Col. 9 Ln. 6 – 12).

63. As to claim 13, Chaudhry teaches to the mutual exclusion mechanism of claim 12, wherein the memory barrier corresponds to a membar operation ("…membar…" Col. 9 Ln. 6 – 12).

64. As to claim 17, Chaudhry teaches the mutual exclusion mechanism of claim 8, wherein the particular read loads lock status (Load Buffer 114 Col. 9 Ln. 1 – 12, Ln. 53 – 67); and wherein the particular write stores a quick lock indication (Store Buffer 116 Col. 9 Ln. 1 – 12, Ln. 53 – 67).

65. As to claim 18, Chaudhry teaches the mutual exclusion mechanism of claim 8, wherein the preclusion is based, at least in part, on characteristics of an implementation of a memory model (Head Thread 202/Speculative Thread 302 Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12).

66. As to claim 51, Chaudhry teaches the lock of claim 48, wherein the acquisition sequence employs a programming construct that precludes reordering of a particular read before a particular write ("...ensure that a write operation occurs before a read operation..." Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

67. As to clam 56, Chaudhry teaches the computer program product of claim 52, wherein the acquisition sequence employs a programming construct that precludes reordering of a particular read before a particular write ("...ensure that a write operation occurs before a read operation..." Col. 8 Ln. 60 – 67, Col. 9 Ln. 1 – 12, Col. 9 Ln. 53 – 67).

68. Claims 9 and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 6,430,649 B1 issued to Chaudhry et al. as applied to claim 8 above, and further in view of U.S. Pat. No. 7,398,376 B2 issued to Mckenney.

69. As to claim 9, Chaudhry and Burrows are silent with reference to the mutual exclusion mechanism of claim 8, wherein the precluded reordering includes reordering by a compiler.

McKenney teaches the mutual exclusion mechanism of claim 8, wherein the precluded reordering includes reordering by a compiler ("…compiler directives…" Abstract).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Chaudhry and Burrows with the teaching of McKenney because teaching of McKenney would improve the system of Chaudhry and Burrows by providing an execution of instruction technique for maximizing processor performance by placing constraints on shared memory access while removing constraints on non-shared memory accesses (McKenney Col. 3 Ln. 35 – 42).

70. As to claim 10, McKenney teaches the mutual exclusion mechanism of claim 8, wherein the precluded reordering includes reordering upon execution by a processor ("…CPU…" Col. 5 Ln. 1 – 3).

71. Claims 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 6,430,649 B1 issued to Chaudhry et al. as applied to claim 8 above, and further in view of U.S. Pat. No. 6,038,646 issued to Sproull.

72. As to claim 14, Chaudhry and Burrows are silent with reference to the mutual exclusion mechanism of claim 12, wherein the memory barrier results from a signal handler.

Sproull teaches the mutual exclusion mechanism of claim 12, wherein the memory barrier results from a signal handler (figure 3(a) Col. 7 Ln. 60 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Chaudhry and Burrows with the teaching of Sproull because teaching of Sproull would improve the system of Chaudhry and Burrows by providing a memory interface barrier that allows a processor to enforce order of concurrently submitted operations, thus optimizing performance (Sproull Col. 4 Ln. 45 – 50).

73. As to claim 15, Sproull teaches the mutual exclusion mechanism of claim 12, wherein the memory barrier results from a cross-call ("…request stream…issued by a processor to a memory subsystem…" Col. 7 Ln. 60 – 67).

74. As to claim 16, Sproull teaches the mutual exclusion mechanism of claim 8, wherein the programming construct is membar-free ("…the processor need not introduce barrier requests to enforce ordering…" Col. 8 Ln. 58 – 61).

75. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. In view of U.S. Pat. No. 6,430,649 B1 issued to Chaudhry et al. as applied to claim 8 above, and further in view of U.S. Pat. No. 6,678,807 B2 issued to Boatright et al.

76. As to claim 19, Chaudhry teaches the mutual exclusion mechanism of claim 8, wherein the programming construct includes use of a store operation followed in program order by a load operation ("…ensure that a write operation occurs before a read operation…" Col. 8 Ln. 62 – 67, Col. 9 Ln. 53 – 59), and Burrows teaches the store and load operations having collocated targets that encode a quick lock indication ("…flag…" Col. 4 Ln. 46 – 48) and lock status, respectively ("…H(M)=0…H(M)=1…" Col. 4 Ln. 1 – 4).

Chaudhry and Burrows are silent with reference to embodying in code compiled for execution on a processor that implements a total store order (TSO) memory model.

Boatright teaches embodying in code compiled for execution on a processor that implements a total store order (TSO) memory model ("…TSO memory model…" Col. 2 Ln. 18 – 38).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Chaudhry and Burrows with the teaching of Boatright because teaching of Boatright would improve the system of Chaudhry and Burrows by providing a mechanism for allowing load operations that are completely covered by two or more store operations to receive data via store buffer forwarding in such a manner as to retain the side effects of the two Total Store Order (TSO) restrictions thereby increasing processor performance without violating the restrictive memory model (Boatright Col. 2 Ln. 7 – 13).

77. Claims 20 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. In view of U.S. Pat. No. 6,965,961 issued to Scott.

78. As to claim 20, Burrows is silent with reference to the mutual exclusion mechanism of claim 1, wherein the biasable lock includes an MCS lock augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed.

Scott teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock includes an MCS lock augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed ("…MCS…" Col. 12 Ln. 15 – 23).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Scott because teaching of Scott would improve the system of Burrows by providing a method of implementing mutual exclusion that is fast, scalable, fair and avoids network contention.

79. As to claim 21, Scott teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock includes an TATAS lock augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed ("…TASB…" Col. 12 Ln. 3 – 23).

80. Claims 22 and 28-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. In view of U.S. Pat. No. 6,112,222 issued to Govindaraju et al.

81. As to claim 22, Burrows is silent with reference to the mutual exclusion mechanism of claim 1, wherein the biasable lock includes a lock provided by a POSIX pthreads mutex library, augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed.

Govindaraju teaches the mutual exclusion mechanism of claim 1, wherein the biasable lock includes a lock provided by a POSIX pthreads mutex library ("…second lock process employing at least one function in the POSIX treads standard…" Col. 2 Ln. 51 – 65), augmented to provide fast path acquisition and release sequences for the thread to which bias has been directed ("…the lock is claimed by the thread and ownership is set…" Col. 3 Ln. 15 – 32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Govindaraju because teaching of Govindaraju would improve the system of Burrows by providing POSIX functions for enhancing portability of applications running across multiple operating systems (Govindaraju Col. 3 Ln. 39 – 41).

82. As to claim 28, Govindaraju teaches the mutual exclusion mechanism of claim 24, further comprising: means for detecting a current level of contention ("…lock state is tested via AIX-provided atomic test functions…" Col. 3 Ln. 15 – 32, Col. 6 Ln. 8 – 13); and a rebiasing sequence that rebiases the lock in response to detection of an absence of contention ("…the lock is claimed by the thread and ownership is set…only one thread is trying to acquire the lock…" Col. 3 Ln. 15 – 32, Col. 6 Ln. 8 – 13).

83. As to claim 29, Govindaraju teaches the mutual exclusion mechanism of claim 28, wherein the contention detection means accesses a request queue to identify the absence of contention ("…count…" Col. 1 – 5).

84. As to claim 30, Govindaraju teaches the mutual exclusion mechanism of claim 1, wherein the contention detection means employs an attempt counter to identify the absence of contention ("…count…" Col. 1 – 5).

85. Claim 27 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. In view of U.S. Pat. No. 6,081,665 A issued to Nilsen et al.

86. As to claim 27, Burrows is silent with reference to the mutual exclusion mechanism of claim 1, wherein revocation of bias by, or on behalf of, a contending thread, is handled, at least in part, at a garbage collection safe point.

Nilsen teaches the mutual exclusion mechanism of claim 1, wherein revocation of bias by, or on behalf of, a contending thread, is handled, at least in part, at a garbage collection safe point ("...point where safe garbage collection can take place..." Col. 66 Ln. 63 – 67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Nilsen because teaching of Nilsen would improve the system of Burrows by providing runtime guarantees that preempted or suspended threads will not make any further progress in execution or use of system resources.

87. Claims 45 and 68-73 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,662,364 B1 issued to Burrows et al. in view of U.S. Pat. No. 6,792,601 B1 issued to Dimpsey et al.

88. As to claim 45, Burrows are silent with reference to the method of claim 36, further comprising: executing the program code in a single-threaded execution environment, wherein the program code is compiled with the biasable lock for execution on both the single-threaded execution environment and a multi-threaded execution environment, and wherein the biasable lock allows the program code to run in the single-threaded execution environment without significant lock-related overhead.

Dimpsey teaches the method of claim 36, further comprising: executing the program code in a single-threaded execution environment ("...first mode..." Col. 6 Ln. 40 – 50, Col. 7 Ln. 26 – 32, Col. 9 Ln. 31 – 39), wherein the program code is compiled with the biasable lock for execution on both the single-threaded execution environment and a multi-threaded execution environment ("...first mode...second mode..." Col. 6 Ln. 40 – 50, Col. 9 Ln. 31 – 39: NOTE: the threads implemented in the first and second modes are Java threads and Java threads/applications are typically compliable before execution), and wherein the lock allows the program code to run in the single-threaded execution environment without significant lock-related overhead ("...object locking system..." Col. 6 Ln. 40 – 47, Col. 7 Ln. 25 – 32).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Dimpsey because teaching of Dimpsey would improve the system of Burrows by providing a mechanism for providing a fast and improved performance by allowing a single thread to use all machine resources.

89. As to claim 68, Burrows teaches a computer program product encoding instructions that implement a biasable lock suitable for execution as a multi-threaded computation ("…multithreaded environment…" Col. 2 Ln. 45 – 50), wherein post-biasing, no atomic operations are executed in the acquisition or release, by a bias-holding thread, of the biasable lock ("…fast nonatomic load/store sequence…" Col. 2 Ln. 53 – 64, "…fast nonatomic synchronization sequence…" Col. 3 Ln. 1 – 3, "…fast nonatomic sequence, H(M)=0…" Col. 4 Ln. 1 – 18, Ln. 56 – 67).

Burrows is silent with reference to implementing a biasable lock suitable for execution as either a single-threaded and a multi-threaded computation.

Dimpsey teaches implementing a lock suitable for execution as either a single-threaded or a multi-threaded computation ("…first mode…second mode…" Col. 6 Ln. 40 – 50, Col. 7 Ln. 26 – 33, Col. 9 Ln. 31 – 39).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Bacon and Burrows with the teaching of Dimpsey because teaching of Dimpsey would improve the system of Bacon and Burrows by providing a mechanism for providing a fast and improved performance by allowing a single thread to use all machine resources.

90. As to claim 69, Dimpsey teaches the computer program product of claim 68, wherein, when executed as the single-threaded computation, the biasable lock imposes minimal lock acquisition and release overhead ("…first mode…" Col. 6 Ln. 40 – 50, Col. 7 Ln. 26 – 33, Col. 9 Ln. 31 – 39).

91. As to claim 70, Burrows teaches the computer program product of claim 68, wherein, when executed as the multi-threaded computation ("…multithreaded environment…" Col. 2 Ln. 45 – 50), the biasable lock imposes minimal lock acquisition and release overhead when repeatedly acquired and released by the bias-holding thread ("…reducing the costs of synchronization in multithreaded environment..." Col. 2 Ln. 45 - 49, "…reducing mutex synchronization overhead…" Col. 3 Ln. 66 – 67).

92. As to claim 71, Burrows teaches the computer program product of claim 68, embodied in at least one computer readable medium selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium (memory 60 Col. 9 Ln. 8 – 20).

93. As to claim 72, Burrows teaches a method of making a single computer program product suitable for efficient execution as a multi-threaded computation ("…multithreaded environment…" Col. 2 Ln. 45 – 50), the method comprising:

structuring a computation as a potentially multithread computation ("…multithreaded environment…" Col. 2 Ln. 45 – 50);

mediating at least some sources of contention in the multithreaded computation using a biasable locking mechanism ("...target mutex..." Col. 2 Ln. 53 - 67, "…Mutex M…" Col. 4 Ln. 1 – 18, Ln. 56 – 67, "…Mutex M (210) Col. 5 Ln. 1 – 36), and introducing the instances of the biasable locking mechanism into program code

("…designate a thread that is currently associated with the mutex…" Col. 2 Ln. 53 – 67, "…the mutex is to be acquired exclusively, or almost exclusively by one thread…" Col. 4 Ln. 7 – 9, Ln. 60 – 61).

Burrows is silent with reference to Burrows is silent with reference to making a single computer program product suitable for efficient execution as a single-threaded computation;

compiling the program code; and

encoding the compiled program code, including the instances of the bias able locking mechanism, in a computer program product.

Dimpsey teaches making a single computer program product suitable for efficient execution as a single-threaded computation ("…first mode…" Col. 6 Ln. 40 – 50, Col. 7 Ln. 26 – 33, Col. 9 Ln. 31 – 39);

compiling the program code (Java Application 50 Col. 9 Ln. 31 – 39: NOTE: the threads implemented in the first and second modes are Java threads and Java threads/applications are typically compliable before execution); and

encoding the compiled program code, including the instances of the bias able locking mechanism (Java Application 50 Col. 9 Ln. 31 – 39), in a computer program product (memory 60 Col. 9 Ln. 8 – 20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Burrows with the teaching of Dimpsey because teaching of Dimpsey would improve the system of Burrows by providing a mechanism for providing a fast and improved performance by allowing a single thread to use all machine resources.

94. As to claim 73, Dimpsey teaches the method of claim 72, wherein the encoding includes transferring the compiled program code onto at least one computer readable medium selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium (memory 60 Col. 9 Ln. 8 – 20).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES E. ANYA whose telephone number is (571)272-3757. The examiner can normally be reached on 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Charles E Anya/
Examiner, Art Unit 2194